

Amendments to the Claims

1 Claim 1 (original): A computer program product embodied on one or more computer-readable
2 media, the computer program product adapted for efficiently transforming extensible structured
3 documents and comprising:

4 computer-readable program code means for identifying a source document type;

5 computer-readable program code means for specifying one or more fast transformations to
6 be performed on documents of the source document type;

7 computer-readable program code means for specifying a source node description and a
8 target node description for each of the specified fast transformations;

9 computer-readable program code means for storing transformation information for each of
10 the specified fast transformations, the transformation information comprising a transformation
11 identifier, the source node description, and the target node description; and

12 computer-readable program code means for processing incoming source documents to
13 generate output documents using the stored transformation information, further comprising:

14 computer-readable program code means for receiving a source document;

15 computer-readable program code means for selecting, manually or based upon a
16 comparison of the received source document to the stored transformation information, zero or
17 more fast transformations to be performed;

18 computer-readable program code means for applying the selected fast
19 transformations; and

20 computer-readable program code means for generating one or more output
21 documents using a result of the computer-readable program code means for applying.

1 Claim 2 (original): The computer program product according to Claim 1, wherein the received
2 source document is an Extensible Markup Language (XML) document.

1 Claim 3 (original): The computer program product according to Claim 2, further comprising
2 computer-readable program code means for parsing the XML document.

1 Claim 4 (original): The computer program product according to Claim 1, wherein the received
2 source document is an array-based representation of an Extensible Markup Language (XML)
3 document.

1 Claim 5 (original): The computer program product according to Claim 4, and wherein the
2 computer-readable program code means for applying the selected transformations further
3 comprises computer-readable program code means for manipulating selected nodes by
4 manipulating the array-based representation.

1 Claim 6 (original): The computer program product according to Claim 1, wherein the received
2 source document is a machine-oriented markup language document.

1 Claim 7 (original): The computer program product according to Claim 1, wherein the received
2 source document is an array-based representation of a machine-oriented markup language
3 document.

1 **Claim 8 (original):** The computer program product according to Claim 1, wherein the received
2 source document is a parsed representation of an extensible document.

1 **Claim 9 (original):** The computer program product according to Claim 1, wherein the source
2 node description identifies one or more source nodes in an input document of the source
3 document type and wherein the target node description identifies zero or more target nodes in an
4 output tree to be generated in the one or more output documents.

1 **Claim 10 (currently amended):** The computer program product according to Claim 1, wherein
2 the computer-readable program code means for processing further comprises computer-readable
3 program code means for applying a general purpose transformation engine to perform
4 transformations other than the selected fast transformations, wherein the general purpose
5 transformation engine is a stylesheet engine, and wherein the computer-readable program code
6 means for generating uses the result of the computer-readable program code means for applying
7 the selected fast transformations as well as a result of the computer-readable program code means
8 for applying the general purpose transformation engine.

1 **Claim 11 (original):** The computer program product according to Claim 10, wherein the
2 stylesheet engine is an Extensible Stylesheet Language (XSL) engine.

1 **Claim 12 (currently amended):** A system for efficiently transforming extensible structured

2 documents, comprising:

3 means for specifying fast transformations to be applied to incoming source documents,

4 further comprising:

5 means for specifying a signature that identifies a source document type;

6 means for specifying one or more fast transformations to be performed on

7 documents that match the specified signature;

8 means for specifying a source node description and a target node description for
9 each of the specified fast transformations; and

10 means for storing transformation information for each of the specified fast
11 transformations, the transformation information comprising a transformation identifier, the source
12 node description, and the target node description;

13 means for applying the fast transformations to particular incoming source documents
14 matching criteria of the specified fast transformations; and

15 means for applying general purpose transformations to incoming source documents not
16 matching criteria of the specified fast transformations.

Claim 13 (canceled)

1 Claim 14 (currently amended): The system according to Claim [[13]] 12, wherein the means for
2 applying the fast transformations further comprises:

3 means for receiving a source document;

4 means for selecting, manually or based upon a comparison of the received source

5 document to the stored transformation information, zero or more fast transformations to be
6 performed; and

7 means for applying the selected fast transformations by manipulating selected nodes of the
8 received source document according to the selected fast transformations.

1 Claim 15 (currently amended): The system according to Claim [[12]] 14, wherein the received
2 source document is an Extensible Markup Language (XML) document.

1 Claim 16 (original): The system according to Claim 15, further comprising means for parsing the
2 XML document.

1 Claim 17 (currently amended): The system according to Claim [[12]] 14, wherein the received
2 source document is an array-based representation of an Extensible Markup Language (XML)
3 document.

1 Claim 18 (original): The system according to Claim 14, wherein the received source document is
2 an array-based representation of an Extensible Markup Language (XML) document, and wherein
3 the means for applying the selected fast transformations by manipulating selected nodes further
4 comprises means for manipulating the array-based representation.

1 Claim 19 (currently amended): The system according to Claim [[12]] 14, wherein the received
2 source document is a machine-oriented markup language document.

1 Claim 20 (currently amended): The system according to Claim [[12]] 14, wherein the received
2 source document is an array-based representation of a machine-oriented markup language
3 document.

1 Claim 21 (currently amended): The system according to Claim [[12]] 14, wherein the received
2 source document is a parsed representation of an extensible document.

1 Claim 22 (currently amended): The system according to Claim 12, wherein the source node
2 description identifies one or more source nodes in an input document of the source document type
3 and wherein the target node description identifies zero or more target nodes in an output tree to
4 be generated in [[the]] one or more output documents, responsive to operation of the means for
5 applying the fast transformations.

1 Claim 23 (currently amended): The system according to Claim 12, wherein the general purpose
2 transformation engine is transformations are applied using a stylesheet engine.

1 Claim 24 (original): The system according to Claim 23, wherein the stylesheet engine is an
2 Extensible Stylesheet Language (XSL) engine.

1 Claim 25 (currently amended): A method for efficiently transforming extensible structured
2 documents, comprising the steps of:

3 specifying fast transformations to be applied to incoming source documents, further
4 comprising the steps of:
5 specifying a signature that identifies a source document type;
6 specifying one or more fast transformations to be performed on documents that
7 match the specified signature;
8 specifying a source node description and a target node description for each of the
9 specified fast transformations; and
10 storing transformation information for each of the specified fast transformations,
11 the transformation information comprising a transformation identifier, the source node
12 description, and the target node description;
13 applying the fast transformations to particular incoming source documents matching
14 criteria of the specified fast transformations; and
15 applying general purpose transformations to incoming source documents not matching
16 criteria of the specified fast transformations.

Claim 26 (canceled)

1 Claim 27 (currently amended): The method according to Claim [[26]] 25, wherein the step of
2 applying the fast transformations further comprises the steps of:
3 receiving a source document;
4 selecting, manually or based upon a comparison of the received source document to the
5 stored transformation information, zero or more fast transformations to be performed; and

6 applying the selected fast transformations by manipulating selected nodes of the received
7 source document according to the selected fast transformations.

1 Claim 28 (currently amended): The method according to Claim [[25]] 27, wherein the received
2 source document is an Extensible Markup Language (XML) document.

1 Claim 29 (original): The method according to Claim 28, further comprising the step of parsing
2 the XML document.

1 Claim 30 (currently amended): The method according to Claim [[25]] 27, wherein the received
2 source document is an array-based representation of an Extensible Markup Language (XML)
3 document.

1 Claim 31 (original): The method according to Claim 27, wherein the received source document
2 is an array-based representation of an Extensible Markup Language (XML) document, and
3 wherein the step of applying the selected fast transformations by manipulating selected nodes
4 further comprises the step of manipulating the array-based representation.

1 Claim 32 (currently amended): The method according to Claim [[25]] 27, wherein the received
2 source document is a machine-oriented markup language document.

1 Claim 33 (currently amended): The method according to Claim [[25]] 27, wherein the received

2 source document is an array-based representation of a machine-oriented markup language
3 document.

1 Claim 34 (currently amended): The method according to Claim [[25]] 27, wherein the received
2 source document is a parsed representation of an extensible document.

1 Claim 35 (currently amended): The method according to Claim 25, wherein the source node
2 description identifies one or more source nodes in an input document of the source document type
3 and wherein the target node description identifies zero or more target nodes ~~in an output tree to~~
4 be generated in [[the]] one or more output documents, responsive to operation of the step of
5 applying the fast transformations.

1 Claim 36 (currently amended): The method according to Claim 25, wherein the general purpose
2 transformation engine ~~is~~ transformations are applied using a stylesheet engine.

1 Claim 37 (original): The method according to Claim 36, wherein the stylesheet engine is an
2 Extensible Stylesheet Language (XSL) engine.

1 Claim 38 (original): The method according to Claim 27, further comprising the step of
2 preloading one or more templates prior to operation of the step of applying the selected fast
3 transformations.

1 Claim 39 (original): The method according to Claim 25, further comprising using a result of the
2 step of applying the fast transformations and a result of the step of applying general purpose
3 transformations to create an output document, and wherein the source document and/or the
4 output document may be represented as in-memory structures which may have been produced by
5 or may be sent to another software process.